About

```
Unsupervised ML algorithms
1) Can work with unlabelled data
2) We do not specify the output
3) Clusters and associates based on features
```

```
--> KMeans Clustering

1. It is a type of unsupervised ML alogrithm
2. Clusters the data points based on the similarities in the features
provided
3. k=No.of clusters
```

```
--> How does KMeans work?

1. Select the value of K (can be random / or based on the elbow method)
    -->Elbow method
        a) plot a graph between no.of clusters and wcss
        b) Choose the point where there is an elbow transition and having
lower wcss value
        c) wcss=within cluster sum of squares
                summation of squares of distances between the points of
each cluster to the centroid of the cluster

2. Random instantiation of centroids for each cluster/Using kMeans++ to
reduce error

3. Assign each datapoint to the cluster where the distance b/w the centroid
of the cluster to the data point is minimal

4. Calculate the centroid of each cluster , considering the members of the
cluster

5. Reassign the datapoints as in step 3

6. If any changes made in clustering, recalculate the centroids, step 4 and
move to step 5.

7. No changes, model ready
```

```
--> Working on data

1) Import the needed libraries
2) Collect the dataset and import it
3) Data preprocessing
4) Extract the needed features
5) Find k value
6) Build the model with found k
```

```
7) predict the cluster
```

In [9]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [10]:

```
data=pd.read_csv("Mall_customers.csv")
```

In [11]:

```
data
```

Out[11]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [12]:

```
data.head()
```

Out[12]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [13]:

```
data.tail()
```

Out[13]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

In [14]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [15]:

```
data.describe()
```

Out[15]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

In [16]:

```
needed_features=data.iloc[:,3:]
```

In [17]:

```
needed_features.head()
```

Out[17]:

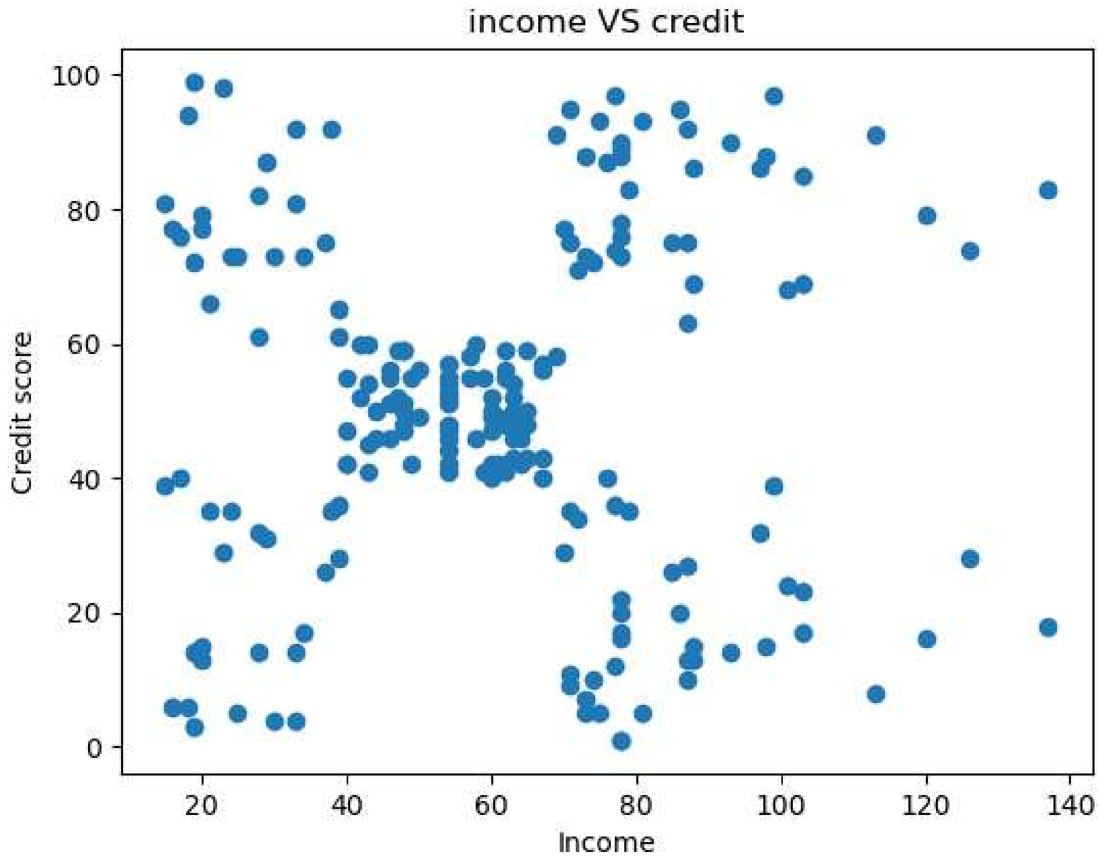
	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

In [18]:

```
x=needed_features.values
```

In [24]:

```
plt.scatter(x[:,0],x[:,1])
plt.title("income VS credit")
plt.xlabel("Income")
plt.ylabel("Credit score")
plt.show()
```



In [32]:

```
wcss=[]

from sklearn.cluster import KMeans

for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init="k-means++",random_state=0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
```

```
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
```

```
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [33]:

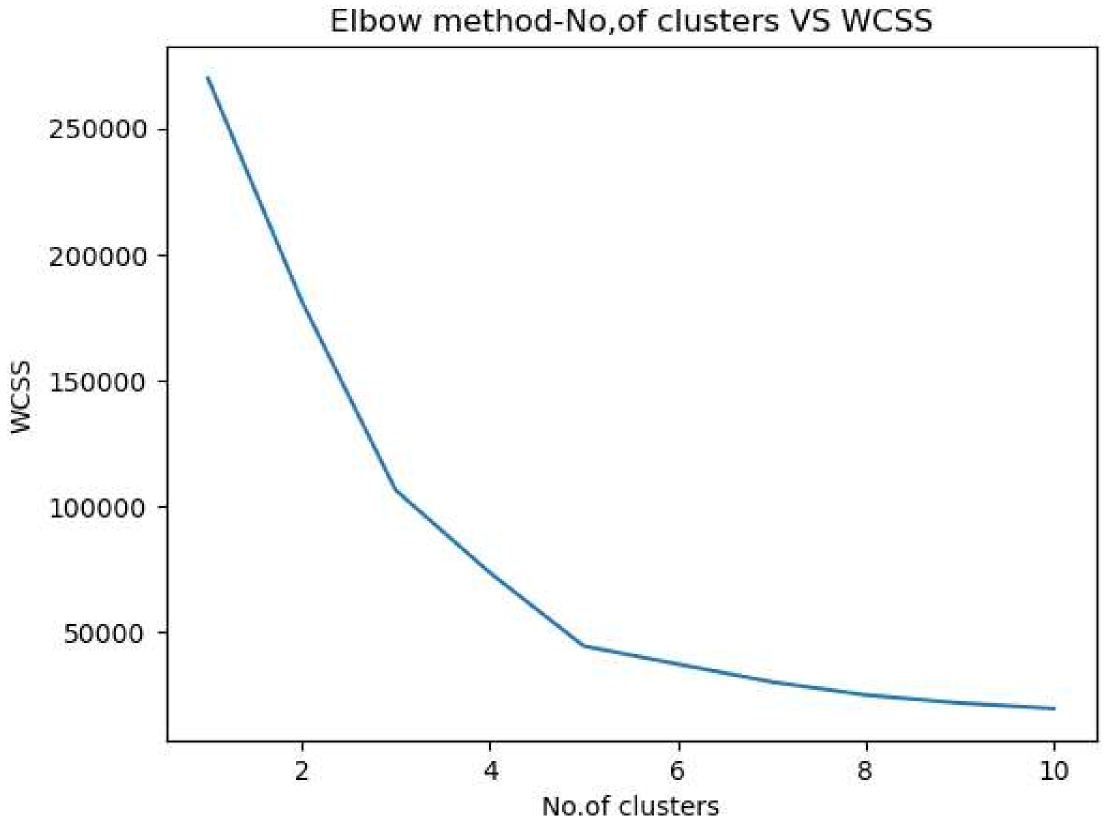
```
wcss
```

Out[33]:

```
[269981.28,
 181363.59595959596,
 106348.37306211119,
 73679.78903948834,
 44448.45544793371,
 37265.86520484346,
 30259.65720728547,
 25050.832307547527,
 21862.09267218289,
 19657.783608703958]
```

In [37]:

```
plt.plot(range(1,11),wcss)
plt.title("Elbow method-No,of clusters VS WCSS")
plt.xlabel("No.of clusters")
plt.ylabel("WCSS")
plt.show()
```



```
Optimal k value=5
```

In [38]:

```
kmeans=KMeans(n_clusters=5,init="k-means++",random_state=0)
```

In [39]:

```
result=kmeans.fit_predict(x)
```

```
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [40]:

```
result
```

Out[40]:

```
array([3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4,
       3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 0,
       3, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 1, 2, 1, 2, 1,
       0, 1, 2, 1, 2, 1, 2, 1, 2, 1, 0, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
       2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
       2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
       2, 1])
```

In [41]:

```
kmeans.cluster_centers_
```

Out[41]:

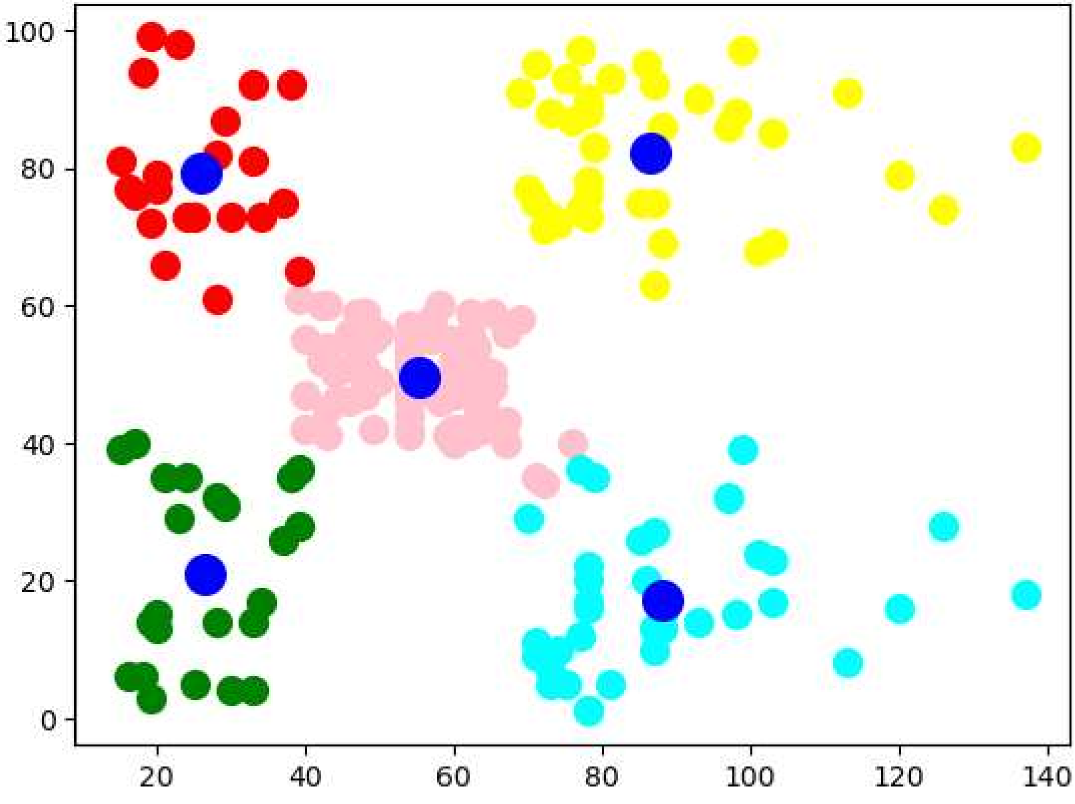
```
array([[55.2962963 , 49.51851852],
       [86.53846154, 82.12820513],
       [88.2       , 17.11428571],
       [26.30434783, 20.91304348],
       [25.72727273, 79.36363636]])
```

In [47]:

```
plt.scatter(x[result==0,0],x[result==0,1],s=100,color='pink',label='cluste
plt.scatter(x[result==1,0],x[result==1,1],s=100,color='yellow',label='clus
plt.scatter(x[result==2,0],x[result==2,1],s=100,color='cyan',label='cluste
plt.scatter(x[result==3,0],x[result==3,1],s=100,color='green',label='clust
plt.scatter(x[result==4,0],x[result==4,1],s=100,color='red',label='cluster
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=20
```

Out[47]:

```
<matplotlib.collections.PathCollection at 0x1bd5554d350>
```



```
--> Clustering
Considering More than 2 features
```

In [48]:

```
x=data.iloc[:,1:]
```

In [49]:

```
x
```

Out[49]:

	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40
...	...	...	...	...
195	Female	35	120	79
196	Female	45	126	28
197	Male	32	126	74
198	Male	32	137	18
199	Male	30	137	83

200 rows × 4 columns

In [64]:

```
# Label Encoding Gender

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
val=x.iloc[:,0]
le.fit(val)
print("Classes = ",le.classes_)
print("After transformation = ",le.transform(val))
```

```
Classes =  ['Female' 'Male']
After transformation =  [1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 0 1 1 0 1
0 1 0 1 0 0 1 0 1 1 0 0 0
 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 1 0 0 1 1 0 0 1 0 1 0 0
0
 1 1 0 1 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1
1
 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 1
0
 0 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0
0
 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1]
```

In [65]:

```
x.iloc[:,0]=le.transform(val)
```

```
C:\Users\TUPAKULA VAISHNAVI\AppData\Local\Temp\ipykernel_273940\18936508
18.py:1: DeprecationWarning: In a future version, `df.iloc[:, i] = newva
ls` will attempt to set the values inplace instead of always setting a n
ew array. To retain the old behavior, use either `df[df.columns[i]] = ne
wvals` or, if columns are non-unique, `df.isetitem(i, newvals)`
  x.iloc[:,0]=le.transform(val)
```

In [66]:

```
x.head()
```

Out[66]:

	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	19	15	39
1	1	21	15	81
2	0	20	16	6
3	0	23	16	77
4	0	31	17	40

In [67]:

```
#finding the value of K

wcss=[]

for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
```

```
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
```

```
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [68]:

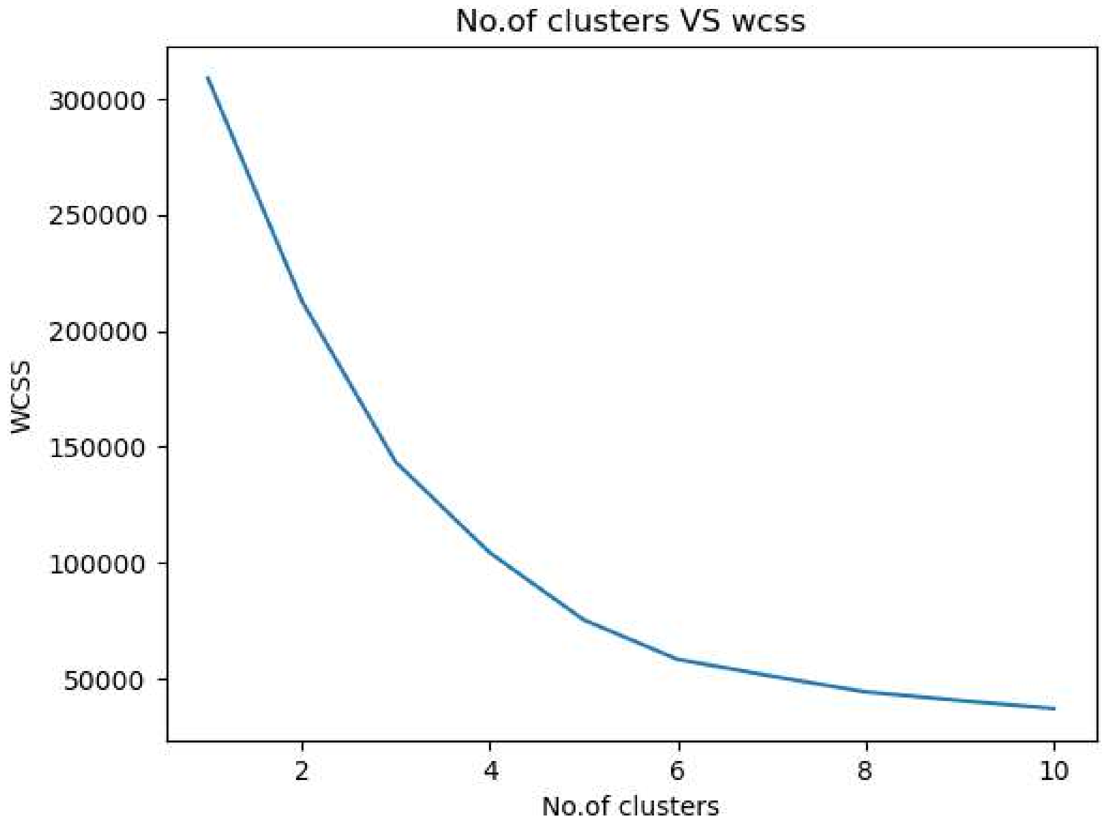
```
wcss
```

Out[68]:

```
[308862.06,
 212889.442455243,
 143391.59236035674,
 104414.67534220166,
 75399.61541401484,
 58348.64136331505,
 51165.184237107926,
 44389.807673557676,
 40670.989962363215,
 37201.52704476877]
```

In [71]:

```
plt.plot(range(1,11),wcss)
plt.title("No.of clusters VS wcss")
plt.xlabel("No.of clusters")
plt.ylabel("WCSS")
plt.show()
```



In [72]:

```
k=6
```

In [73]:

```
kmeans=KMeans(n_clusters=6,init="k-means++",random_state=0)
ymeans=kmeans.fit_predict(x)
```

```
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1412: FutureWarning: The default value of `n_init` will chan
ge from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to sup
press the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\TUPAKULA VAISHNAVI\anaconda3\Lib\site-packages\sklearn\cluster
\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You
can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
In [74]:  ymeans
```

```
Out[74]: array([4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5,
                4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 3, 5, 3, 0,
                4, 5, 3, 0, 0, 0, 3, 0, 0, 3, 3, 3, 3, 3, 0, 3, 3, 0, 3, 3, 3, 0,
                3, 3, 0, 0, 3, 3, 3, 3, 3, 0, 3, 0, 0, 3, 3, 0, 3, 3, 0, 3, 3, 0,
                0, 3, 3, 0, 3, 0, 0, 0, 3, 0, 3, 0, 0, 3, 3, 0, 3, 0, 3, 3, 3, 3,
                3, 0, 0, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 2, 0, 2, 1, 2, 1, 2, 1, 2,
                0, 2, 1, 2, 1, 2, 1, 2, 1, 2, 0, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
                1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
                1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
                1, 2])
```

```
In [78]:  kmeans.cluster_centers_
```

```
Out[78]: array([[ 0.34210526, 27.        , 56.65789474, 49.13157895],
                [ 0.57142857, 41.68571429, 88.22857143, 17.28571429],
                [ 0.46153846, 32.69230769, 86.53846154, 82.12820513],
                [ 0.44444444, 56.15555556, 53.37777778, 49.08888889],
                [ 0.38095238, 44.14285714, 25.14285714, 19.52380952],
                [ 0.40909091, 25.27272727, 25.72727273, 79.36363636]])
```